



How to enable Android (AOSP) on your developer board



Mattijs Korpershoek <mkorpershoek@baylibre.com>

\$ whoami

- Mattijs (/ma'tɛjs/) Korpershoek 
- Toulouse, France
- Embedded Android/Linux engineer, BayLibre
 - embedded software consultancy based in Nice, France, with ~60 engineers around the world contributing to open source projects like Linux, U-Boot, Android, Yocto and Zephyr
- U-Boot custodian (u-boot-dfu, usb gadget, fastboot, android AB/AVB)



Contents

[1- Introduction](#)

[2- Getting started with Android](#)

[3- Android Common Kernel \(ACK\)](#)

[4- Bootloaders](#)

[5- Booting Android to home screen](#)

[6- Thank you & questions](#)



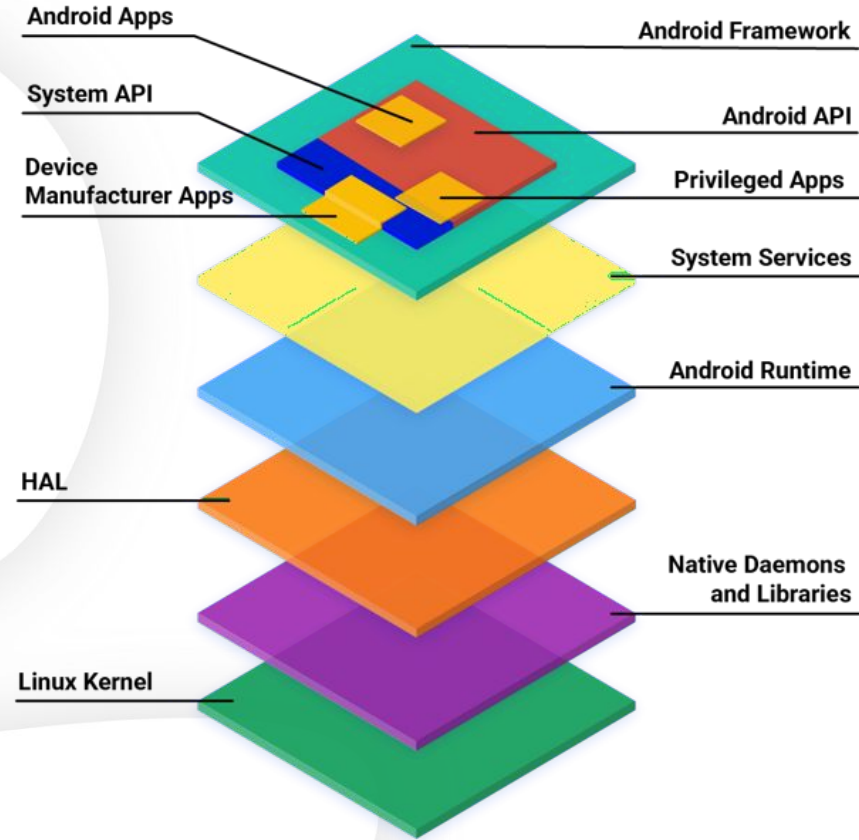
1- Introduction: why port Android ?

- To use your computer as a heater in cold winters
- Feature rich operating system
- Robust OTA system
- Well known [developer API](#) for apps
- Lots of developers available to make custom UI/UX for Android
- For learning purposes



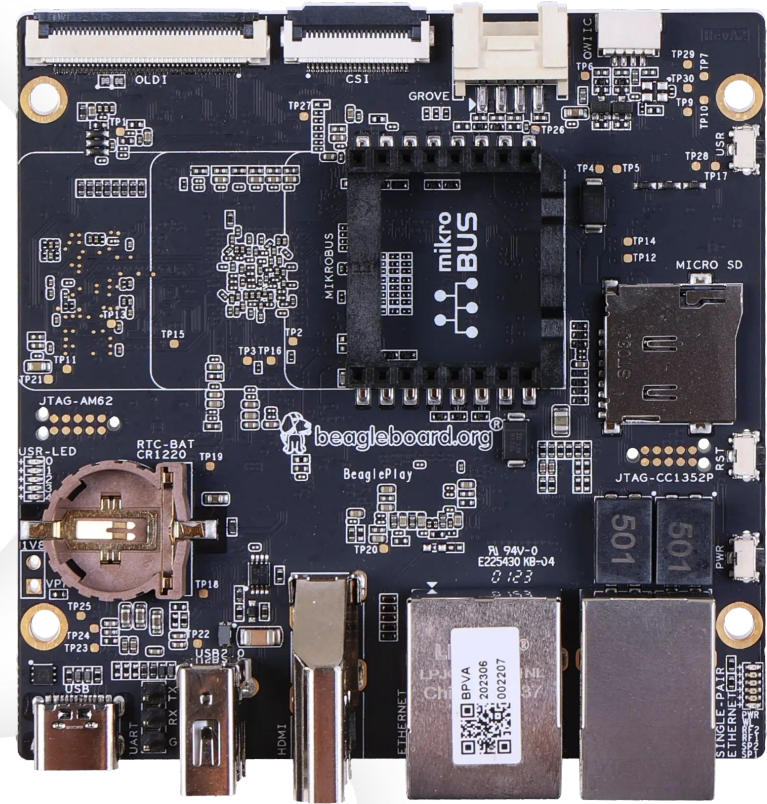
1- Android overview

- **HAL** (Hardware Abstraction Layer)
- Native Daemons and Libraries
- Linux Kernel



1- The developer board

- BeaglePlay (beagleboard.org)
- Texas Instruments Sitara **AM62X**
- 1.4GHz quad-core Arm Cortex-A53
- **2GB** DDR4
- **16GB** eMMC flash
- HDMI out
- USB-C
- UART debug



1- The plan

- Download the sources
- Load kernel from bootloader
- Boot to console
- Boot to adb devices
- Boot to boot animation
- Boot to home screen



1- Code alert



Attention
Source code

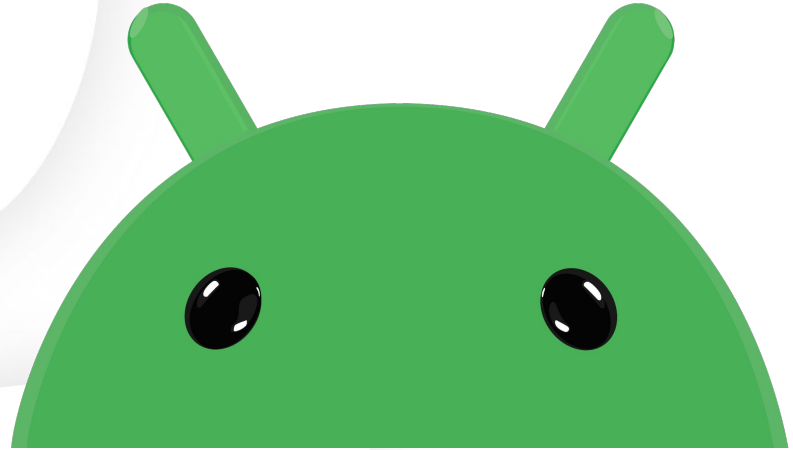


2- Getting started with Android



2- Android: getting the source

- source.android.com
- **A**ndroid **O**pen **S**ource **P**roject (AOSP)
- branch/tag
 - main
 - android-14.0.0_r60



2- Android: getting the source

- Repo: python wrapper around git
- **1355** git projects in aosp/android-14.0.0_r60
- **222 GB** of source code
- Build machine requirements (**64 GB** of ram recommended)

```
$ MANIFEST=https://android.googlesource.com/platform/manifest
$ repo init -u $MANIFEST -b android-14.0.0_r60
$ repo sync -cd -j5

[...]
Fetching: 100% (1355/1355), done in 36m18.594s
[...]
Checking out: 100% (1355/1355), done in 5m34.490s
repo sync has finished successfully.
```



2- Android: source tree overview

```
$ ls
Android.bp      build      development  kernel      packages    system      WORKSPACE
art             BUILD     device      libcore     pdk         test
bionic         cts       external    libnativehelper platform_testing toolchain
bootable       dalvik    frameworks  lk_inc.mk   prebuilts   tools
bootstrap.bash developers hardware    out         sdk         trusty
```

- **device**: board support
- **hardware**: HAL (Hardware Abstraction Layer)
- **external**: Upstream projects that are imported to AOSP
- **system**: system services (init, adb, logs, ...)



2- Android: build



```
$ source build/envsetup.sh  
$ lunch
```

```
You're building on Linux  
Warning: Cannot display lunch menu.
```



2- Android: build

```
$ lunch aosp_cf_x86_64_phone-trunk_staging-userdebug
build/make/core/release_config.mk:145: error: No release config found for
TARGET_RELEASE: trunk_staging. Available releases are: ap2a.
10:29:19 dumpvars failed with: exit status 1
```

```
$ lunch aosp_cf_x86_64_phone-ap2a-userdebug
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=14
PRODUCT_INCLUDE_TAGS=com.android.mainline
mainline_module_prebuilt_nightly
TARGET_PRODUCT=aosp_cf_x86_64_phone
TARGET_BUILD_VARIANT=userdebug
[...]
OUT_DIR=out
```



2- Android build: CuttleFish

- Virtual Android device
- Reference code
- Used for real-world testing



```
$ rg --type amake aosp_cf_x86_64_phone
```

```
device/google/cuttlefish/AndroidProducts.mk
```

```
37:      aosp_cf_x86_64_phone:$(LOCAL_DIR)/vsoc_x86_64/phone/aosp_cf.mk \
```

```
38:      aosp_cf_x86_64_phone_vendor:$(LOCAL_DIR)/vsoc_x86_64/phone/aosp_cf_vendor.mk
```



2- Android build: beagleplay lunch target

```
$ mkdir -p device/beagle/beagleplay  
$ cd device/beagle/beagleplay/  
$ touch AndroidProduct.mk  
$ touch product.mk
```

AndroidProduct.mk

```
PRODUCT_MAKEFILES := \  
    beagleplay:${LOCAL_DIR}/product.mk  
  
COMMON_LUNCH_CHOICES := \  
    beagleplay-ap2a-userdebug
```

```
$ lunch beagleplay-ap2a-userdebug
```

```
build/make/core/board_config.mk:231: error: No config file found for TARGET_DEVICE  
beagleplay.
```



2- Android build: beagleplay: board definitions

BoardConfig.mk

```
TARGET_BOARD_PLATFORM := beagleplay
TARGET_ARCH := arm64
TARGET_ARCH_VARIANT := armv8-a
TARGET_CPU_ABI := arm64-v8a
TARGET_CPU_VARIANT := cortex-a53
```

product.mk

```
PRODUCT_NAME := BeaglePlay
PRODUCT_DEVICE := beagleplay
PRODUCT_MANUFACTURER := BeagleBoard
PRODUCT_MODEL := BeaglePlay

PRODUCT_BRAND := Android
```



2- Android build: lunch & make



```
$ lunch beagleplay-ap2a-userdebug
```

```
PLATFORM_VERSION=14
```

```
TARGET_PRODUCT=beagleplay
```

```
TARGET_BUILD_VARIANT=userdebug
```

```
TARGET_ARCH=arm64
```

```
TARGET_ARCH_VARIANT=armv8-a
```

```
TARGET_CPU_VARIANT=cortex-a53
```

```
BUILD_ID=AP2A.240805.005.E1
```

```
OUT_DIR=out
```

```
$ m
```

```
## build completed successfully (13:05 (mm:ss)) ##
```



2- Android build: output directory

- Where are the images ?
- Multiple partitions
- <https://source.android.com/docs/core/architecture/partitions>

```
$ ls out/target/product/beagleplay
```

all_modules.txt	fake_packages	misc_info.txt	ramdisk.img
android-info.txt	fastboot-info.txt	module-info.json	ramdisk_node_list
build_fingerprint.txt	installed-files-ramdisk.json	module-info.json.rsp	root
build_thumbprint.txt	installed-files-ramdisk.txt	obj	symbols
clean_steps.mk	installed-files-root.json	previous_build_config.mk	system
debug_ramdisk	installed-files-root.txt	ramdisk	



2- Android: possible partition scheme

misc	Bootloader Control Block
boot_{a,b}	Kernel and generic ramdisk
vendor_boot_{a,b}	Vendor specific ramdisk
super	Dynamic partitions
metadata	File metadata encryption data
userdata	Encrypted user files



2- Android: A/B Updates & Dynamic Partitions

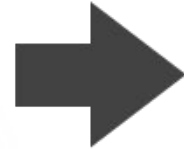
boot_{a,b}

vendor_boot_{a,b}

- A/B system updates
- **Slots:** 2 sets of partitions:
 - boot_a
 - boot_b
- Current vs. Unused slot
- Rollback

super

- Userspace partitions
- GPT: super
- /system
- /vendor
- Virtual A/B
- Resize
- physical
- logical



2- Android: partitions: BoardConfig

misc
boot_{a,b}
vendor_boot_{a,b}
super
metadata
userdata

BoardConfig.mk

```
# Partitions
BOARD_BOOTIMAGE_PARTITION_SIZE := 67108864 # 64MiB

BOARD_USES_METADATA_PARTITION := true

BOARD_USERDATAIMAGE_PARTITION_SIZE := 442450944 # 6GiB
BOARD_USERDATAIMAGE_FILE_SYSTEM_TYPE := f2fs

# Super partition (Dynamic Partitions)
BOARD_SUPER_PARTITION_SIZE := 4294967296 # 4 GiB
BOARD_BUILD_SUPER_IMAGE_BY_DEFAULT := true
# [...]
```



2- Android: let's rebuild

\$ m

FAILED: ninja: 'out/target/product/beagleplay/kernel', needed by 'out/target/product/beagleplay/boot.img', missing and no known rule to make it

11:45:04 ninja failed with: exit status 1

- Kernel prebuilts ?
- Where to download ?
- Android kernel versions ?

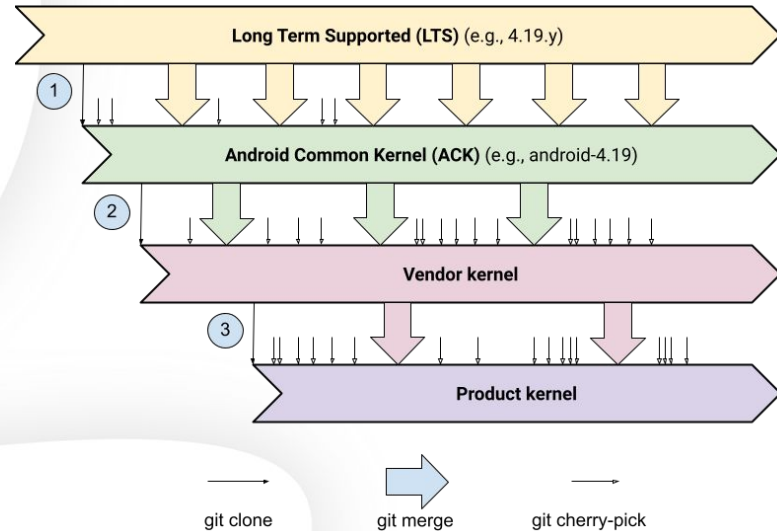


3- Android Common Kernel (ACK)



3- Android Common Kernel (ACK)

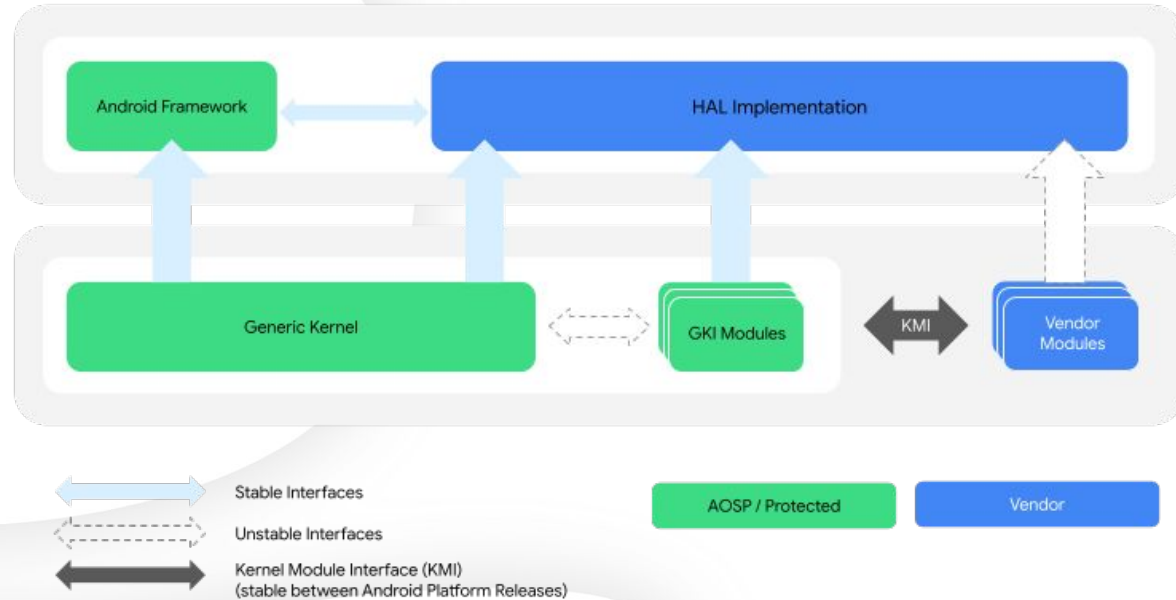
- Downstream of kernel.org kernels
- Versions requirements per Android release
- Android-14:
 - **android14-6.1**
 - **android14-5.15**
- Fetched via repo
- Outside of Android source tree
- Generic Kernel Image ([GKI](#))
- Take most recent: **android14-6.1**



3- Generic Kernel Image (GKI)

- Kernel fragmentation
- Device kernels
- Update issues

- Single kernel **Image**
- Vendor **modules**
- drivers must work as modules

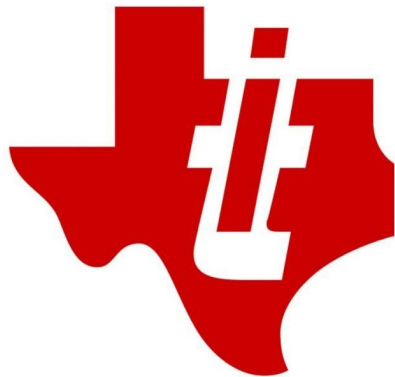


3- Android kernel: backporting from 6.4



- **Problem:** Beagle Play support landed in 6.4
- **android14-6.1**

- Texas Instruments public [ti-linux-kernel](#) tree
- Device tree
- Drivers: UART, CLK, SPI, I2C, watchdog, MMC, PMIC, ...
- **1784** patches later ...



- Android [porting rules](#): UPSTREAM:, BACKPORT:, FROMGIT:, FROMLIST:, ANDROID:



3- Android kernel: build system

- [Kleaf documentation](#)
- BUILD.bazel
- Bazel macros
- GKI defconfig
- CONFIG_ARCH_K3
- defconfig fragment
- [ABI symbol list](#)

```
# edit gki_defconfig
$ tools/bazel run //common:kernel_aarch64_config \
  -- menuconfig

# build for beagleplay
$ tools/bazel build //common:beagleplay

# build and copy artifacts to "dist" directory
$ tools/bazel run //common:beagleplay_dist -- \
  --dist_dir=dist
```



3- Android kernel: build system

- BUILD.bazel
- Add modules to list
- Built-in (=y)
- Module (=m)
- Driver behaviour

```
_BEAGLEPLAY_MODULE_OUTS = ["drivers/tty/serial/8250/8250_omap.ko", ]

kernel_build(
    name = "beagleplay",
    outs = [
        "Image",
        # [...]
        "k3-am625-beagleplay.dtb",
        # [...]
    ],
    build_config = "build.config.beagleplay",
    collect_unstripped_modules = True,
    kmi_symbol_list = "android/abi_gki_aarch64_beagleplay",
    make_goals = [
        "Image",
        "modules",
        "ti/k3-am625-beagleplay.dtb",
    ],
    module_outs = get_gki_modules_list("arm64") + _BEAGLEPLAY_MODULE_OUTS,
    strip_modules = True,
)
```



3- Android kernel: add to Android build system

- Copy binaries to Android: device/beagle/beagleplay-kernel/6.1
- Modules for vendor ramdisk

BoardConfig.mk

```
TARGET_KERNEL_USE := 6.1
TARGET_KERNEL_PATH := device/beagle/beagleplay-kernel/$(TARGET_KERNEL_USE)
KERNEL_MODULES_PATH := $(TARGET_KERNEL_PATH)

PRODUCT_COPY_FILES += $(TARGET_KERNEL_PATH)/Image:kernel

BOARD_VENDOR_RAMDISK_KERNEL_MODULES := \
    $(KERNEL_MODULES_PATH)/8250_omap.ko \
    # more modules needed in ramdisk
```



3- Android kernel: device tree binary

- Build system logic handled by Android
- k3-am625-beagleplay.dtb
- End of vendor_boot partition

BoardConfig.mk

```
# dtb at end of vendor_boot.img  
BOARD_PREBUILT_DTBIMAGE_DIR := $(TARGET_KERNEL_PATH)  
BOARD_INCLUDE_DTB_IN_BOOTIMG := true
```



3- Android: back to building boot images

```
$ m bootimage vendorbootimage
```

```
build/make/core/config.mk:798: warning: This device does not have Treble enabled. This is unsafe.
```

```
#### build completed successfully (03:16 (mm:ss)) ####
```

```
$ file out/target/product/beagleplay/boot.img
```

```
out/target/product/beagleplay/boot.img: Android bootimg, kernel (0x1973c5), ramdisk (0x630)
```

- Treble? Unsafe ?



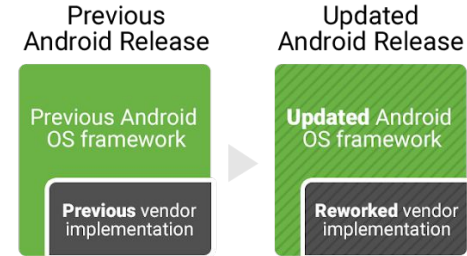
3- Android: Treble and Shipping API

- Treble: Android 8
- PRODUCT_SHIPPING_API_LEVEL
- [Build codenames](#)
- Android 14: API level **34**

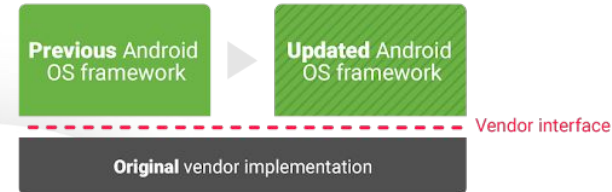
product.mk

PRODUCT_SHIPPING_API_LEVEL := **34**

Before Treble



With Treble



3- Android: ramdisk content

- Init (not systemd)
 - Empty ?
 - Android should provide core packages
-
- Cuttlefish
 - PRODUCT_PACKAGES
 - inherit-product

```
$ tree out/target/product/beagleplay/ramdisk/  
ramdisk  
├── debug_ramdisk  
├── dev  
├── metadata  
├── mnt  
├── proc  
├── second_stage_resources  
├── sys  
└── system  
    ├── etc  
    └── ramdisk  
        └── build.prop
```



3- Android: inheritance

```
$ get_build_var PRODUCT_PACKAGES | sed 's/ /\n/g' | wc -l  
0
```

product.mk

```
$(call inherit-product, $(SRC_TARGET_DIR)/product/full_base.mk)
```

```
$ get_build_var PRODUCT_PACKAGES | sed 's/ /\n/g' | wc -l  
757
```



3- Android: init in ramdisk

- `init`
- Time to try to boot our images

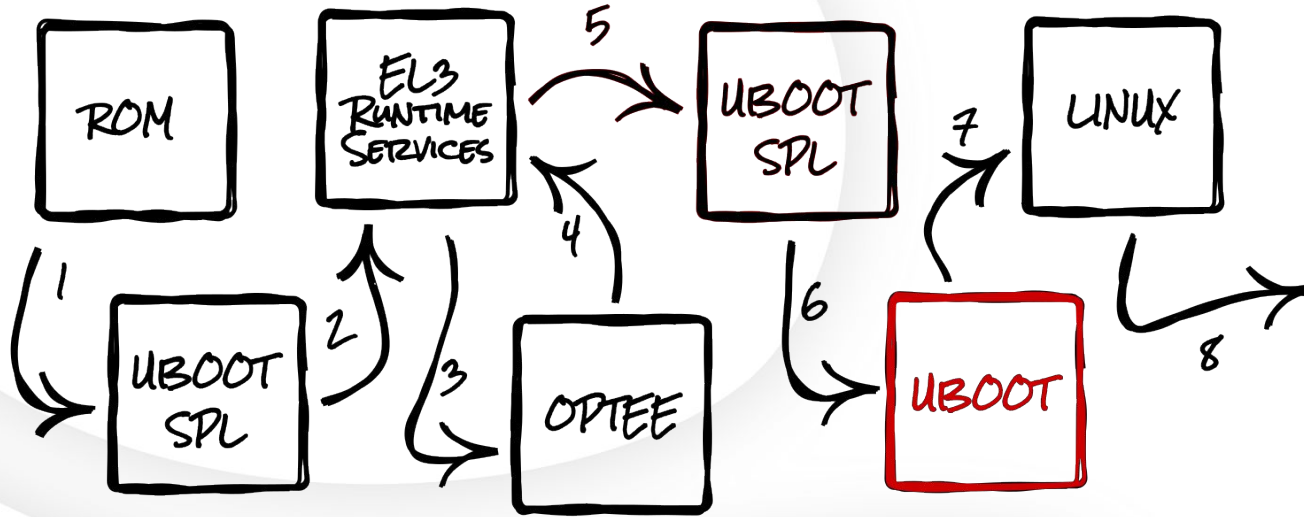
```
$ tree out/target/product/beagleplay/ramdisk/  
ramdisk  
├── debug_ramdisk  
├── dev  
├── init  
├── metadata  
├── mnt  
├── proc  
├── second_stage_resources  
├── sys  
└── system  
    ├── etc  
    │   └── ramdisk  
    │       └── build.prop
```



4- Bootloaders



4- Bootloaders: AM62X bootflow



4- Bootloaders: U-Boot

- No version requirements from Android
 - Vendor specific implementation
 - [Android bootloader requirements](#)
-
- Pick a recent version: **U-Boot v2024.10-rc3**
 - Beagle Play: great [upstream documentation](#)
 - Booting Android: supported via [Android Bootmeth](#)

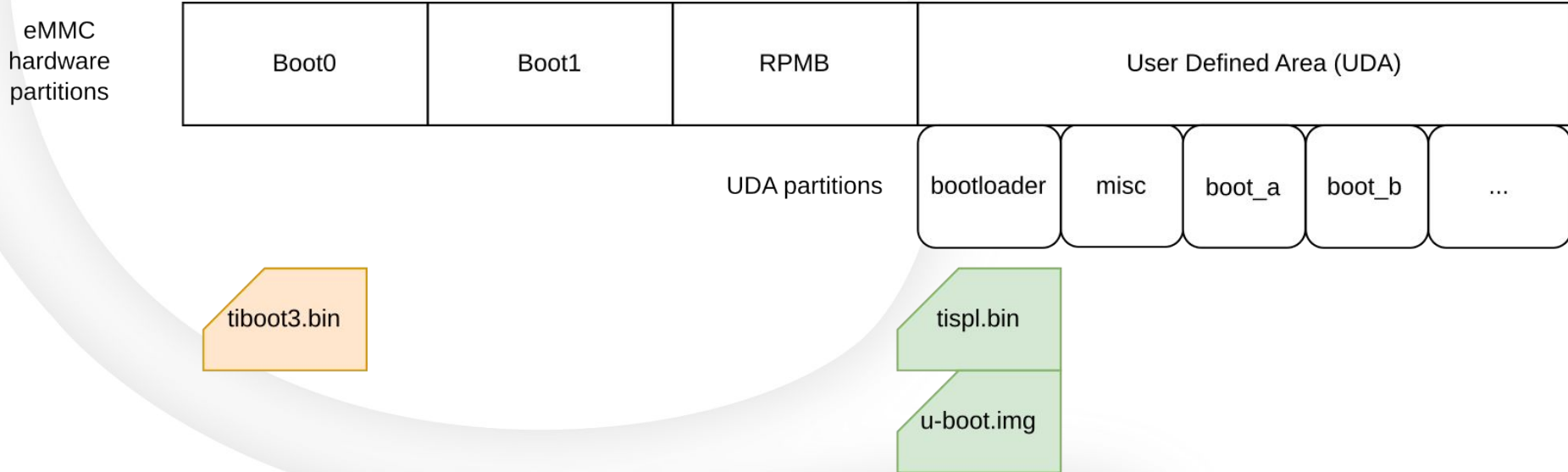


4- Bootloaders: U-Boot

- Enable defconfig options
 - CONFIG_USB_FUNCTION_FASTBOOT
 - CONFIG_BOOTMETH_ANDROID
- Add partitions to `beagleplay.env`
- Change boot medium to `mmcdev=0`
- Changes [submitted upstream](#) by Guillaume La Roque



4- Bootloaders: eMMC layout



- Flash on eMMC via fastboot



4- Bootloaders: bootflow command

```
=> bootflow scan -l
```

```
Scanning for bootflows in all bootdevs
```

Seq	Method	State	Uclass	Part	Name	Filename
---	-----	-----	-----	----	-----	-----

```
Scanning bootdev 'mmc@fa10000.bootdev':
```

```
ANDROID: Invalid CRC-32 (expected 807077e9, found 00000000), re-initializing A/B metadata.
```

```
ANDROID: Attempting slot a, tries remaining 7
```

0	android	ready	mmc	0	<u>mmc@fa10000.bootdev.whole</u>	
----------	----------------	-------	-----	----------	----------------------------------	--

```
No more bootdevs
```

```
(1 bootflow, 1 valid)
```

```
=> bootflow select 0
```

```
=> bootflow boot
```



4- Bootloaders: bootflow boot

=> bootflow boot

** Booting bootflow 'mmc@fa10000.bootdev.whole' with android

ANDROID: Attempting slot a, tries remaining 7

Booting Android Image at 0x82000000 ...

Kernel load addr 0x82001000 size 34915 KiB

Kernel extra command line:

"Error" handler, esr 0xbf000000

elr: 00000000808a5fa8 lr : 0000000080807be8 (reloc)

elr: 00000000fff67fa8 lr : 00000000ffec9be8

x0 : 0000000011000000 x1 : 00000000d0000800

x2 : 00000000000041bc x3 : 0000000000000080

x4 : 0000000000000000 x5 : 00000000fff8af3f

[...]

Code: d65f03c0 f8636824 f8236804 91002063 (cb030044)

Resetting CPU ...



4- Android: mkbootimg / kernel cmdline

- Bootloader expectations
- mkbootimg arguments
- Kernel commandline

BoardConfig.mk

```
BOARD_KERNEL_OFFSET := 0x82000000
BOARD_RAMDISK_OFFSET := 0xd0000000
BOARD_MKBOOTIMG_ARGS += --kernel_offset $(BOARD_KERNEL_OFFSET)
BOARD_MKBOOTIMG_ARGS += --ramdisk_offset $(BOARD_RAMDISK_OFFSET)
BOARD_MKBOOTIMG_ARGS += --pagesize 4096
BOARD_MKBOOTIMG_ARGS += --header_version 4

BOARD_KERNEL_CMDLINE += init=/init
BOARD_KERNEL_CMDLINE += console=ttyS2,115200
BOARD_KERNEL_CMDLINE += 8250.nr_uarts=10
BOARD_KERNEL_CMDLINE += printk.devkmsg=on
```



4- Android: kernel boots

```
## Booting Android Image at 0x82000000 ...
```

```
Kernel load addr 0x92000000 size 34915 KiB
```

```
Kernel extra command line: printk.devkmsg=on init=/init console=ttyS2,115200 8250.nr_uarts=10
```

```
Working FDT set to d0006000
```

```
Loading Kernel Image to 92000000
```

```
Starting kernel ...
```

```
[...]
[ 0.000000][ T0] Booting Linux on physical CPU 0x0000000000 [0x410fd034]
```

```
[ 0.000000][ T0] Machine model: BeagleBoard.org BeaglePlay
```

```
[...]
[ 5.091116][ T1] init: [libfstab] ReadDefaultFstab(): failed to find device default fstab
[ 5.099612][ T1] init: Failed to create FirstStageMount : failed to read default fstab for first stage mount
[ 5.109882][ T1] init: FirstStageMount not available
[ 5.116448][ T1] init: InitFatalReboot: signal 6
```



5- Booting Android to home screen



5- Android: mounting partitions

- “FirstStageMount not available”
- fstab: list mount points
- First stage mount
- Second stage mount
- androidboot.hardware

BoardConfig.mk

```
BOARD_BOOTCONFIG := androidboot.boot_devices=bus@f0000/fa10000.mmc
BOARD_BOOTCONFIG += androidboot.hardware=beagleplay
PRODUCT_COPY_FILES += \
    device/beagle/beagleplay/fstab:$(TARGET_COPY_OUT_VENDOR_RAMDISK)/first_stage_ramdisk/fstab.beagleplay
```

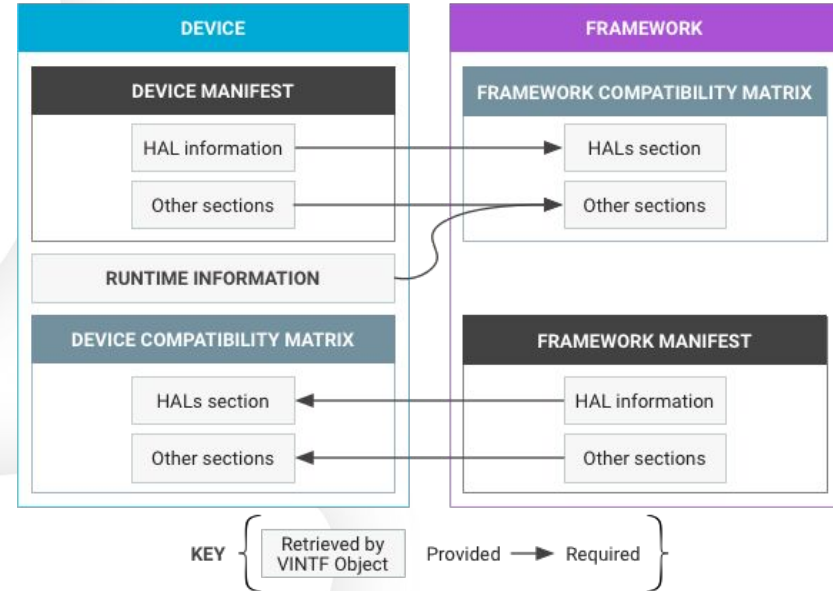


5- Android: time to build the other images

\$ m

ERROR: No such device: No **device manifest** file from device or from update package

- DEVICE_MANIFEST_FILE
- Hardware Abstraction Layer (HAL)
- Features exposed to Android system
- Add an empty one to continue build



5- Android: mounting system

```
[ 5.453749] init: [libfs_mgr] __mount(source=/dev/block/dm-0,target=/system,type=ext4)=0:  
          Success  
[ 5.463940] init: Switching root to '/system'
```

- First stage mount ✓
- Second stage mount ?
- logcat

```
console:/ $ su  
console:/ # dmesg -n1  
console:/ # logcat
```



5- Android: keystore

```
init: starting service 'keystore2'...
```

```
init: ... started service 'keystore2' has pid 199
```

```
DEBUG: Abort message: 'Failed to create service android.system.keystore2.IKeystoreService/default'  
# [...]
```

```
DEBUG:      1: system/security/keystore2/src/globals.rs:339: Cannot connect to Keymint
```

```
DEBUG:      2: system/security/keystore2/src/globals.rs:250: Trying to get Legacy wrapper.
```

```
DEBUG:      3: Error::Km(r#HARDWARE_TYPE_UNAVAILABLE).
```

- System: Keystore
- Vendor: Keymint / keymaster HAL, Gatekeeper HAL
- Encryption/authentication: **mandatory for userdata**
- Legacy software implementation (insecure)



5- Android: add Keymaster HAL to image

- HALs are vendor specific
- Vendor image
- Device manifest
- Init and SEPolicy context
- Permissive for bringup

```
<hal format="hidl">
<name>android.hardware.keymaster</name>
<transport>hwbinder</transport>
<version>3.0</version>
<interface>
  <name>IKeymasterDevice</name>
  <instance>default</instance>
</interface>
</hal>
```


```
u:object_r:hal_gatekeeper_default_exec:s0
u:object_r:hal_keymaster_default_exec:s0
```

product.mk

```
PRODUCT_PACKAGES += \
    android.hardware.keymaster@3.0-service \
    android.hardware.keymaster@3.0-impl \
    android.hardware.gatekeeper@1.0-service.software
```



5- Android: userdata encryption

- File metadata (size, permissions)
- Mandatory for Android 11+
- Protected by Keymaster
- Init.rc
- Second stage mount 

init.rc

```
on early-fs
    start vold

on fs
    mount_all /vendor/etc/fstab.${ro.hardware} --early

on late-fs
    mount_all /vendor/etc/fstab.${ro.hardware} --late
```

[13.609408][T1] init: **Userdata mounted** using /vendor/etc/fstab.beagleplay



5- Android Debug Bridge (ADB)

- USB debugging
- Pulling/pushing files, logcat
- configs (deprecated)
- Cuttlefish (no usb configs)
- Dragonboard
- Dedicated init.rc
- Serialnumber

\$ adb devices

List of devices attached
beagleplay1234 device

init.usb.beagleplay.rc

```
on boot
    mount configs none /config
    mkdir /config/usb_gadget/g1 0770 shell shell
    mkdir /config/usb_gadget/g1/strings/0x409 0770 shell shell
    write /config/usb_gadget/g1/bcdUSB 0x0200
    # [...]
```



5- Android: graphics rendering

F DEBUG : Abort message: 'couldn't find an OpenGL ES implementation, make sure one of persist.graphics.egl, ro.hardware.egl and ro.board.platform is set'

- SwiftShader
- Rendering on the CPU
- Cuttlefish
- Dragonboard

product.mk

```
$(call inherit-product, device/linaro/dragonboard/shared/graphics/swangle/device.mk)
```



5- Android: display

- external/minigbm
- android.graphics allocator
- android.graphics.mapper
- TI Display SubSystem (TIDSS)
- /dev/dri/card1
- VKMS: virtual display
- external/drm_hwcomposer
- android.graphics.hwcomposer



5- Android: missing bits

- Audio HAL: yukawa
- Power HAL: AOSP example
- Health HAL: AOSP example

- Device manifest
- SELinux context

product.mk

```
PRODUCT_PACKAGES += \  
    android.hardware.audio.service \  
    android.hardware.audio@6.0-impl \  
    android.hardware.audio.effect@6.0-impl \  
    audio.primary.$(TARGET_BOARD_PLATFORM)
```

```
PRODUCT_PACKAGES += \  
    android.hardware.power-service.example
```

```
PRODUCT_PACKAGES += \  
    android.hardware.health-service.example \  
    android.hardware.health-service.example_recovery
```



5- Android: keep awake

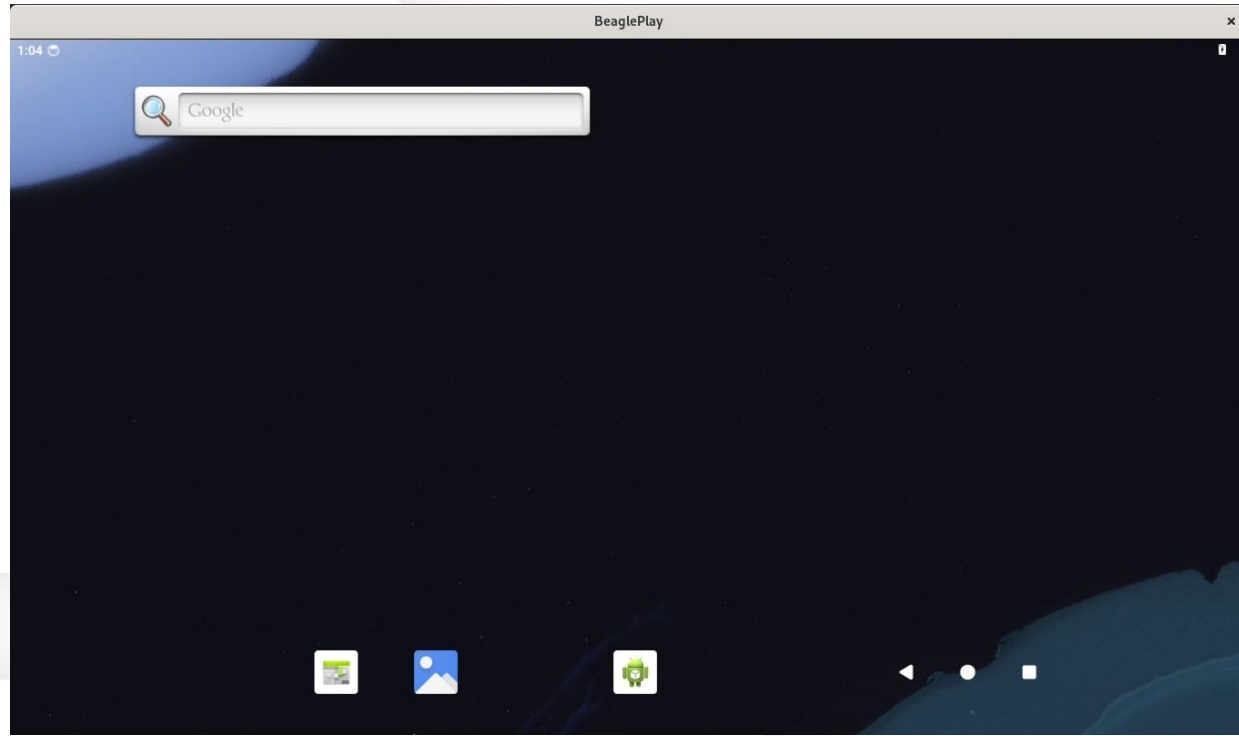
- Avoid board going into suspend
- Talk to power manager via adb

```
$ adb shell svc power stayon true
```



5- Android: the home screen

```
$ scrcpy --no-audio
```



<https://github.com/Genymobile/scrcpy>

5- Android: To sum up

- fstab
- Keymaster and gatekeeper
- userdata and metadata encryption
- ADB via init.usb.beagleplay.rc
- Graphics via Swiftshader
- Display via minigbm and drm_hwcomposer
- Audio, power and health HALs

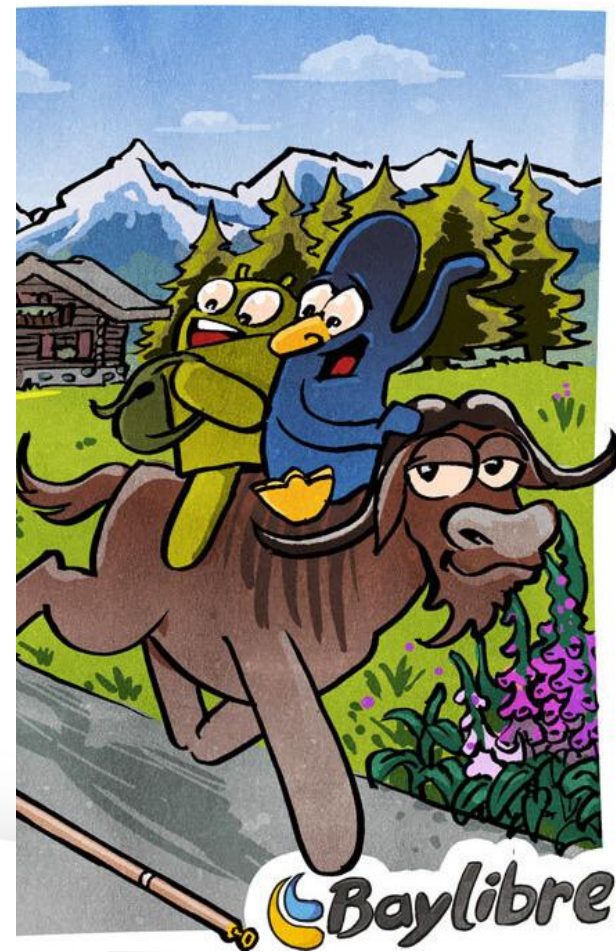


5- Android: what's next

- Very slow: not ready for production
- Enforcing SELinux
- Graphics with GPU (mesa, vendor libs)
- Display support
- Security via OP-TEE
- Connectivity
- Recovery
- Camera

Code available at:

<https://gitlab.baylibre.com/mkorpershoek/beagleplay-aosp>



Thank you
Questions ?



Credits

- Most illustrations come from Frank Tizzoni
 - <https://frank-tizzoni.com/wp/index.php>



Backup: Bootloaders: flashing to eMMC

- Beagle Play supports [USB DFU](#) via USB button
- Different config (am62x_r5_usbdfu.config), make sure to use it
- Via dfu-util or [snagboot](#)
- Once U-Boot starts, we can flash with fastboot

```
$ fastboot flash tiboot3 tiboot3.bin
$ IMAGE=bootloader-beagleplay.img
$ dd if=/dev/zero of="$IMAGE" bs=1048576 count=8 status=none
$ mkfs.vfat "$IMAGE"
$ mcopy -i "$IMAGE" tisp1.bin ::tisp1.bin
$ mcopy -i "$IMAGE" u-boot.img ::u-boot.img
$ fastboot flash bootloader bootloader-beagleplay.img
```



Backup: flashing Android to eMMC

```
$ cd out/target/product/beagleplay
$ fastboot erase misc
$ fastboot flash boot boot.img && \
  fastboot flash vendor_boot vendor_boot.img
$ fastboot flash super super.img
$ fastboot flash userdata userdata.img
$ fastboot format metadata
$ fastboot reboot
```

- Erase `misc` to reset the slot selection

